“Our Chat” – A Chatting Website

A Project Testing Report

Submitted by

Chirag Shrivastava(N272) **Divyansh Tiwari (N283)**

Prateek Shukla (N273) **Shalaka Avahad (N294)**

Under the Guidance of

Dr. Gaurav Londhe

in partial fulfillment for the award of the degree of

MBA (Tech.)

COMPUTER ENGINEERING

At

SVKM'S NMIMS Deemed-to-be UNIVERSITY

MUKESH PATEL SCHOOL OF TECHNOLOGY MANAGEMENT & ENGINEERING

October 2021

ACKNOWLEDGEMENT

We would like to express our gratitude to Prof. Gaurav Londhe who guided us throughout the phase of testing our project “**Our Chat”.** He helped us in the thorough understanding of fundamentals of project testing, understanding the basics testing frameworks, implementing the testing frameworks and making our project more responsive and much efficient.

Contents

INTRODUCTION

PROJECT OVERVIEW

In our project, one can do live chat with others. For this have created a web interface which is directly connected to a SQL Server which stores the user data and his connections and related chat data. This connection is established using PHP mysqli and is called/ retrieved to the client side by sending regular Ajax Requests.

The user account data is stored in SQL server with a unique ID, and then he/she can connect with other users on the platform and add them as a friend. This new relationship creates a new Table in the SQL server which will be used to store their chatting history. Whenever a new message is sent in, that is stored as a new entry in this table. The web client sends regular Ajax requests to the server to check for any new messages and if a new message is present it is displayed on the screen.

BACKGROUND AND MOTIVATON

Software testing is the process of evaluating and verifying that a software product or application does what it is supposed to do. Human errors can cause a defect or failure at any stage of the software development life cycle. The results are classified as trivial or catastrophic, depending on the consequences of the error.

The requirement of rigorous testing and their associated documentation during the software development life cycle arises because of the below reasons:

- To identify defects
- To reduce flaws in the component or system
- Increase the overall quality of the system

There can also be a requirement to perform software testing to comply with legal requirements or industry-specific standards. These standards and rules can specify what kind of techniques should we use for product development. For example, the motor, avionics, medical, and pharmaceutical industries, etc., all have standards covering the testing of the product.

OBJECTIVE

The points below shows the significance of testing for a reliable and easy to use software product:

- The testing is important since it discovers defects/bugs before the delivery to the client, which guarantees the quality of the software.
- It makes the software more reliable and easy to use.
- Thoroughly tested software ensures reliable and high-performance software operation.

METHODOLOGY

We used Tidy to make the code more efficient. Tidy is a console application for macOS, Linux, Windows, UNIX, and more. It corrects and cleans up HTML and XML documents by fixing markup errors and upgrading legacy code to modern standards. Selenium was used with selenium framework for unit testing of the project.

ABOUT THE TOOLS

1. *TIDY*

There are four basic principles to a tidy API:

i. *Reuse existing data structures.*

Where possible, re-use existing data structures, rather than creating custom data structures for your own package. Generally, it's better to prefer common existing data structures over custom data structures, even if slightly ill-fitting.

ii. *Compose simple functions with the pipe.*

A powerful strategy for solving complex problems is to combine many simple pieces. Each piece should be easily understood in isolation, and have a standard way to combine with other pieces. In R, this strategy plays out by composing single functions with the pipe. The pipe, %>%, is a common composition tool that works across all packages.

iii. *Embrace functional programming.*

R is a functional programming language; embrace it, don't fight it. If you're familiar with an object-oriented language like Python or C#, this is going to take some adjustment. But in the long run you will be much better off working with the language, rather than fighting it.

iv. *Design for humans.*

Designing the API primarily so that it is easy to use by humans. Computer efficiency is a secondary concern because the bottleneck in most data analysis is thinking time, not computing time.

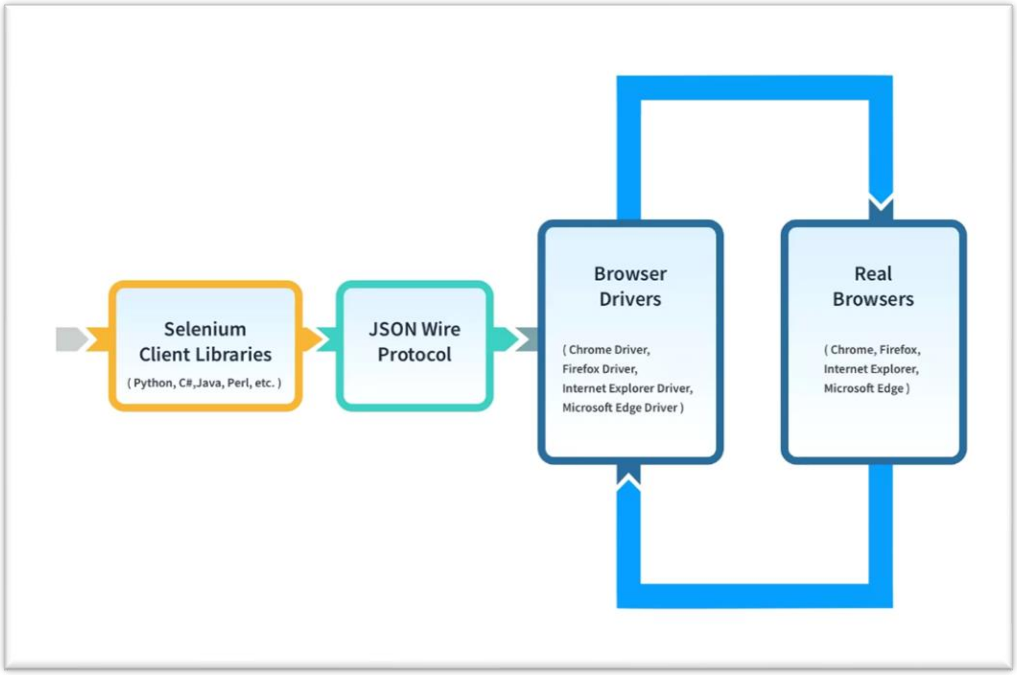
2. *SELENIUM*

Selenium is a powerful tool for controlling web browsers through programs and performing browser automation. It permits you to execute cross-browser tests. It is functional for all browsers, works on all major OS and its scripts are written in various languages i.e. Python, Java, C#, etc.,

We worked with Python using Selenium WebDriver and WebElement to perform Unit Testing with selenium. Selenium WebDriver allows you to choose a programming language to create test scripts.

WebDriver Architecture is made up of four major components:

1. Selenium Client library
2. JSON wire protocol over HTTP
3. Browser Drivers
4. Browsers

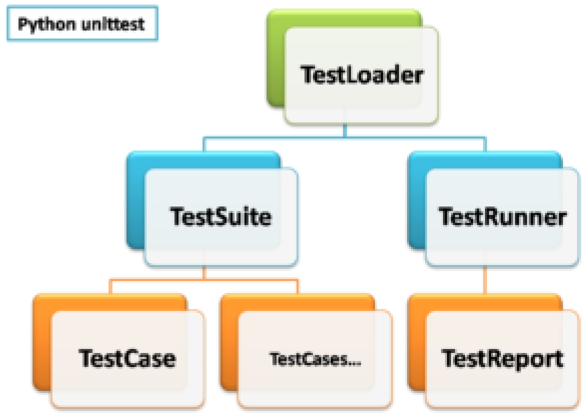

Selenium Client Libraries
(Python, C#,Java, Perl, etc.)
JSON Wire Protocol
Browser Drivers
(Chrome Driver, Firefox Driver, Internet Explorer Driver, Microsoft Edge Driver)
Real Browsers
(Chrome, Firefox, Internet Explorer, Microsoft Edge)

3. *PYTHON Unittest Framework*

Python Unittest library inherits its root from a third-party module known as PyUnit. It was Steve Purcell who ideated PyUnit based on the famous JUnit framework. And later it grew as an official Python module beginning from version 2.5.

Like the JUnit, Python Unittest module splits up its functionality among five key components. All five elements work in tandem to support automation testing. Let's discuss each of them one by one in detail.

Architecture:



Test Loader – It's a Python class which loads test cases and suites created locally or from an external data source like a file. It releases a TestSuite object that carries those cases and suites.

Test Case – The TestCase class holds the test handlers and provides hooks for preparing each handler and for cleaning up after execution.

Test Suite – It acts as a container for grouping test cases. With the help of a test suite, you can combine a set of test cases representing specific functionalities of the application under test.

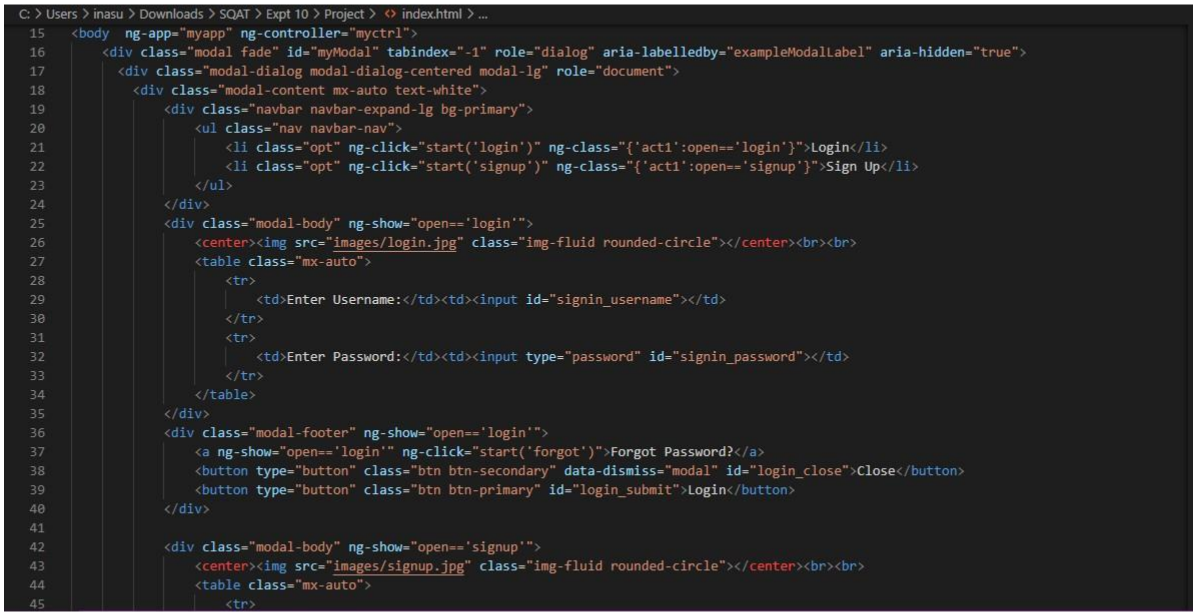
Test Runner – It provides a runnable interface for the execution of tests and delivers the results to the user. It can use channels like a GUI, a textual medium, or return a standard code to notify the results of test execution.

Test Report – This component organizes test results, display pass/fail status of the executed test cases. It even provides the details of steps, a summary of overall run and the time lapsed in execution.

IMPLEMENTATION

RUNNING TIDY ON OUR PROJECT

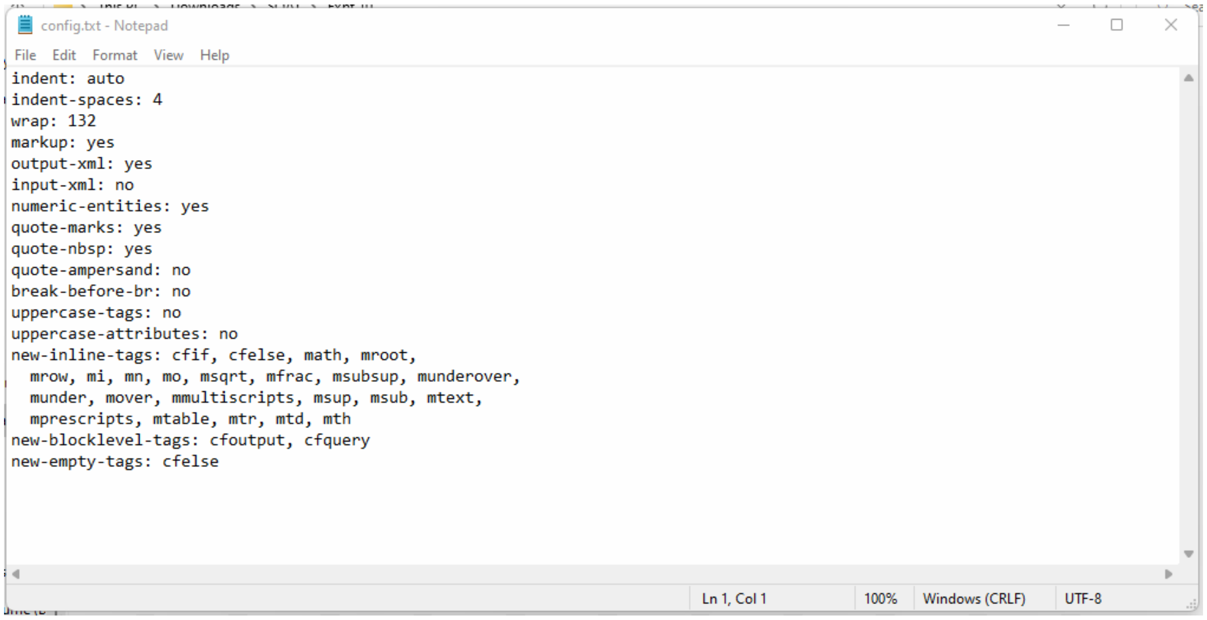
This is how our code file initially looked like, before running TIDY on it



```
C: > Users > inasu > Downloads > SQAT > Expt 10 > Project > <> index.html > ...
<body  ng-app="myapp" ng-controller="myctrl">
    <div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">
      <div class="modal-dialog modal-dialog-centered modal-lg" role="document">
        <div class="modal-content mx-auto text-white">
            <div class="navbar navbar-expand-lg bg-primary">
                <ul class="nav navbar-nav">
                    <li class="opt" ng-click="start('login')" ng-class="{'act1':open=='login'}">Login</li>
                    <li class="opt" ng-click="start('signup')" ng-class="{'act1':open=='signup'}">Sign Up</li>
                </ul>
            </div>
            <div class="modal-body" ng-show="open=='login'">
                <center><img src="images/login.jpg" class="img-fluid rounded-circle"></center><br><br>
                <table class="mx-auto">
                    <tr>
                        <td>Enter Username:</td><td><input id="signin_username"></td>
                    </tr>
                    <tr>
                        <td>Enter Password:</td><td><input type="password" id="signin_password"></td>
                    </tr>
                </table>
            </div>
            <div class="modal-footer" ng-show="open=='login'">
                <a ng-show="open=='login'" ng-click="start('forgot')">Forgot Password?</a>
                <button type="button" class="btn btn-secondary" data-dismiss="modal" id="login_close">Close</button>
                <button type="button" class="btn btn-primary" id="login_submit">Login</button>
            </div>

            <div class="modal-body" ng-show="open=='signup'">
                <center><img src="images/signup.jpg" class="img-fluid rounded-circle"></center><br><br>
                <table class="mx-auto">
                    <tr>
```

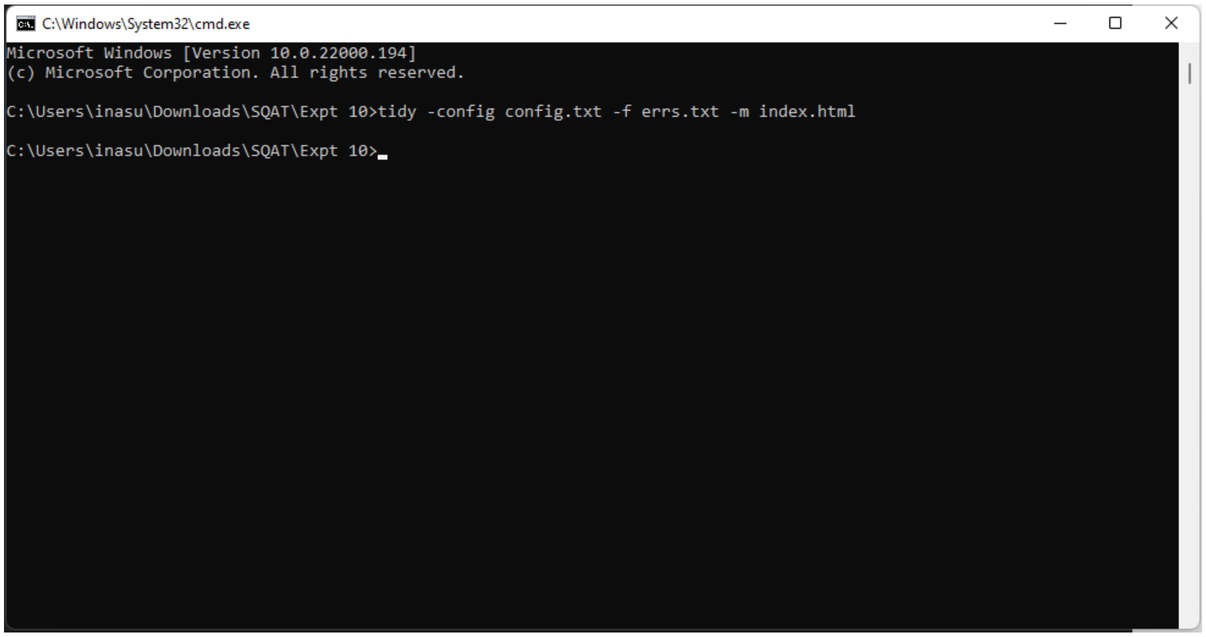
We'll be running TIDY on this file to check for incorrect code formatting and then fix it using TIDY as well.

We can specify our customized Formatting needs to TIDY when executing it using a config file. This is the config file we created:



```
indent: auto
indent-spaces: 4
wrap: 132
markup: yes
output-xml: yes
input-xml: no
numeric-entities: yes
quote-marks: yes
quote-nbsp: yes
quote-ampersand: no
break-before-br: no
uppercase-tags: no
uppercase-attributes: no
new-inline-tags: cfif, cfelse, math, mroot,
  mrow, mi, mn, mo, msqrt, mfrac, msubsup, munderover,
  munder, mover, mmultiscripts, msup, msub, mtext,
  mprescripts, mtable, mtr, mtd, mth
new-blocklevel-tags: cfoutput, cfquery
new-empty-tags: cfelse
```

Running TIDY on our code file, we are using CMD to run it:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.194]
(c) Microsoft Corporation. All rights reserved.

C:\Users\inasu\Downloads\SQAT\Expt 10>tidy -config config.txt -f errs.txt -m index.html

C:\Users\inasu\Downloads\SQAT\Expt 10>
```

Output generated:

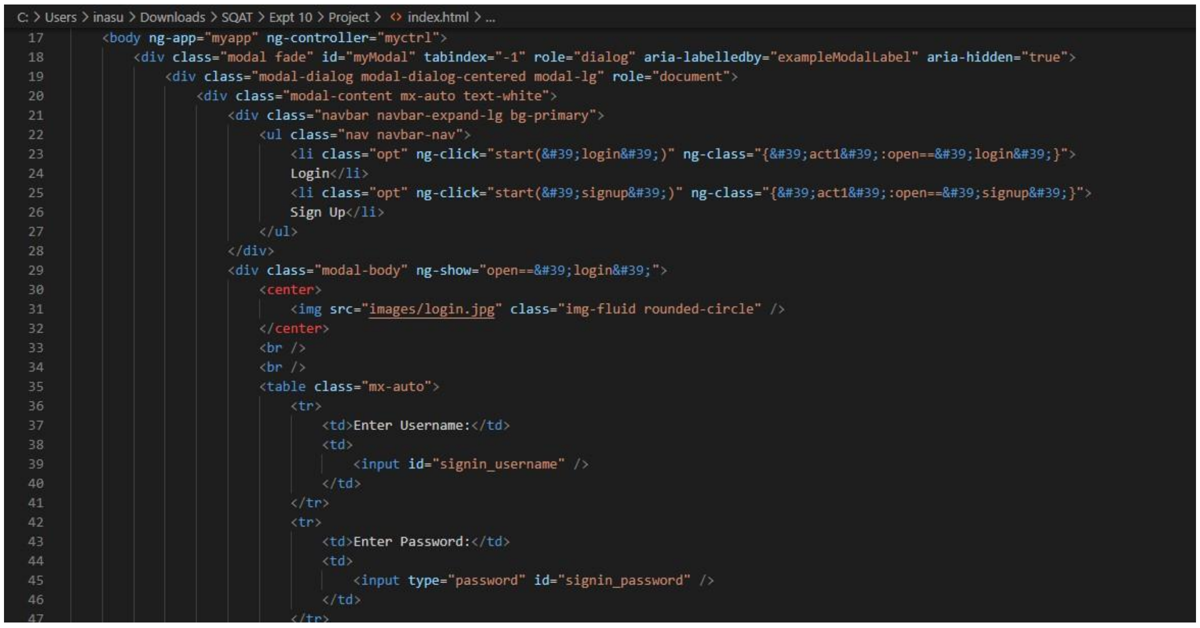
```
line 29 column 9 - Warning: <img> lacks "alt" attribute
line 33 column 9 - Warning: <table> lacks "summary" attribute
line 54 column 9 - Warning: <img> lacks "alt" attribute
line 58 column 9 - Warning: <table> lacks "summary" attribute
line 96 column 9 - Warning: <img> lacks "alt" attribute
line 100 column 9 - Warning: <table> lacks "summary" attribute
line 137 column 9 - Warning: <img> lacks "alt" attribute
line 141 column 9 - Warning: <table> lacks "summary" attribute
line 149 column 9 - Warning: <input> anchor "dob" already defined
line 178 column 9 - Warning: <img> lacks "alt" attribute
line 182 column 9 - Warning: <table> lacks "summary" attribute
line 28 column 9 - Warning: <center> element removed from HTML5
line 53 column 9 - Warning: <center> element removed from HTML5
line 95 column 9 - Warning: <center> element removed from HTML5
line 136 column 9 - Warning: <center> element removed from HTML5
line 177 column 9 - Warning: <center> element removed from HTML5
line 217 column 9 - Warning: <center> element removed from HTML5
line 17 column 9 - Warning: <body> proprietary attribute "ng-app"
line 17 column 9 - Warning: <body> proprietary attribute "ng-controller"
line 23 column 9 - Warning: <li> proprietary attribute "ng-click"
line 23 column 9 - Warning: <li> proprietary attribute "ng-class"
line 24 column 9 – Warning: <li> proprietary attribute ″ng-click″
line 24 column 9 – Warning: <li> proprietary attribute ″ng-class″
line 27 column 9 – Warning: <div> proprietary attribute ″ng-show″
line 48 column 9 – Warning: <div> proprietary attribute ″ng-show″
line 49 column 9 – Warning: <a> proprietary attribute ″ng-show″
line 49 column 9 – Warning: <a> proprietary attribute ″ng-click″
line 52 column 9 – Warning: <div> proprietary attribute ″ng-show″
line 91 column 9 – Warning: <div> proprietary attribute ″ng-show″
line 94 column 9 – Warning: <div> proprietary attribute ″ng-show″
line 109 column 9 – Warning: <div> proprietary attribute ″ng-show″
 line 236 column 9 – Warning: <div> proprietary attribute ″ng-hide″
line 243 column 9 – Warning: <div> proprietary attribute ″ng-show″
line 244 column 9 – Warning: <a> proprietary attribute ″ng-class″
line 244 column 9 – Warning: <a> proprietary attribute ″ng-modal″
line 244 column 9 – Warning: <a> proprietary attribute ″ng-click″
line 244 column 9 – Warning: <a> proprietary attribute ″ng-repeat″
line 249 column 9 – Warning: <div> proprietary attribute ″ng-hide″
line 258 column 9 – Warning: <div> proprietary attribute ″ng-show″
line 258 column 9 – Warning: <div> proprietary attribute ″ng-repeat″
line 263 column 9 – Warning: <div> proprietary attribute ″ng-repeat″
line 266 column 9 – Warning: <span> proprietary attribute ″ng-show″
Info: Document content looks like HTML Proprietary
Tidy found 42 warnings and 0 errors!

The table summary attribute should be used to describe the table structure. It is very helpful
for people using non-visual browsers.
The scope and headers attributes for table cells are useful for specifying which headers apply
to each table cell, enabling non-visual browsers to provide a meaningful context for each cell.

The alt attribute should be used to give a short description of an image; longer descriptions
should be given with the longdesc attribute which takes a URL linked to the description.
These measures are needed for people using non-graphical browsers.
```

For further advice on how to make your pages accessible see http://www.w3.org/WAI/GL.
About HTML Tidy: https://github.com/htacg/tidy-html5
Bug reports and comments: https://github.com/htacg/tidy-html5/issues Official mailing list: https://lists.w3.org/Archives/Public/public-htacg/ Latest HTML specification: http://dev.w3.org/html5/spec-author-view/ Validate your HTML documents: http://validator.w3.org/nu/
Lobby your company to join the W3C: http://www.w3.org/Consortium

Do you speak a language other than English, or a different variant of English? Consider helping us to localize HTML Tidy. For details please see https://github.com/htacg/tidy-html5/blob/master/README/LOCALIZE.md

For the code file TIFY returned 42 warnings and 0 Errors. While most of the errors are associated to AngularJS CDN, there are some warnings associated to the "Accessibility" of the website. We need to add "alt" attributes to images and "summary" tags to the tables.

TIDY has also corrected the formatting for us. This is the Thus outputted File:



C: > Users > inasu > Downloads > SQAT > Expt 10 > Project > index.html > ...

```html
    <body ng-app="myapp" ng-controller="myctrl">
        <div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">
            <div class="modal-dialog modal-dialog-centered modal-lg" role="document">
                <div class="modal-content mx-auto text-white">
                    <div class="navbar navbar-expand-lg bg-primary">
                        <ul class="nav navbar-nav">
                            <li class="opt" ng-click="start(&#39;login&#39;)" ng-class="{&#39;act1&#39;:open==&#39;login&#39;}">
                            Login</li>
                            <li class="opt" ng-click="start(&#39;signup&#39;)" ng-class="{&#39;act1&#39;:open==&#39;signup&#39;}">
                            Sign Up</li>
                        </ul>
                    </div>
                    <div class="modal-body" ng-show="open==&#39;login&#39;">
                        <center>
                            <img src="images/login.jpg" class="img-fluid rounded-circle" />
                        </center>
                        <br />
                        <br />
                        <table class="mx-auto">
                            <tr>
                                <td>Enter Username:</td>
                                <td>
                                    <input id="signin_username" />
                                </td>
                            </tr>
                            <tr>
                                <td>Enter Password:</td>
                                <td>
                                    <input type="password" id="signin_password" />
                                </td>
                            </tr>
```

UNIT TESTING USING SELENIUM ON PYTHON

Tool: Selenium WebDriver
Programming Language: Python

We can divide our project into 3 Modules and in this part we'll be performing testing on each module separately and ensure each module is working fine.

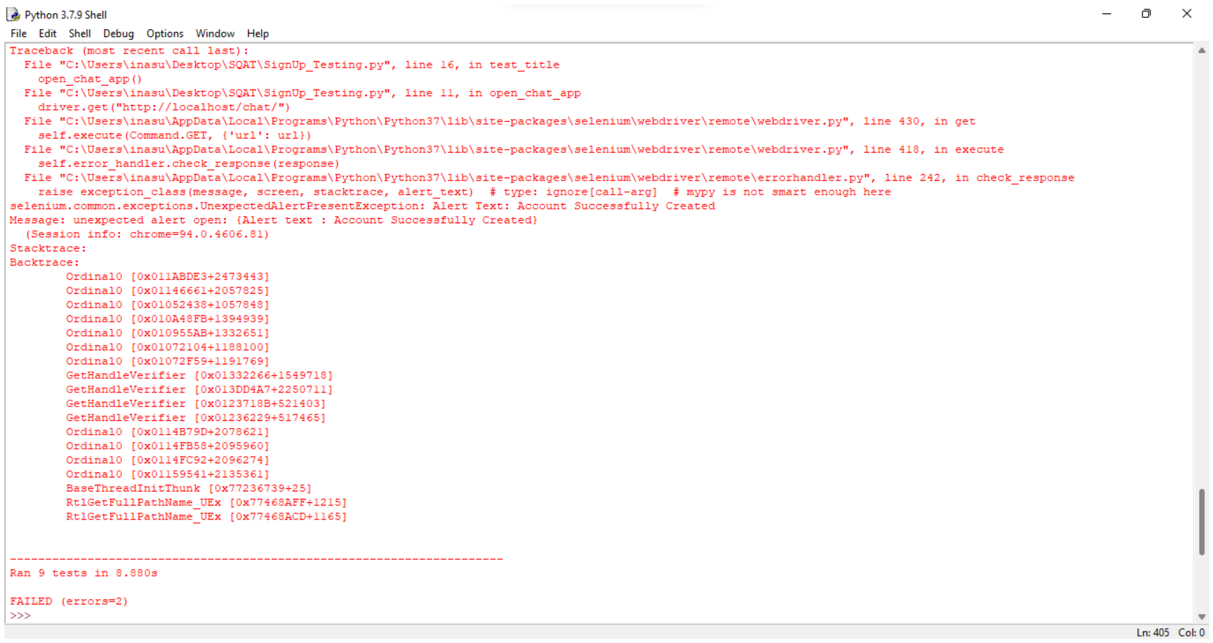
Module 1: Sign-up

For the signup module we need to check that All the details (First and last name, DOB, email) are entered correctly. Both the passwords entered match and none of the fields are empty. If any of these conditions are not satisfied an appropriate error should be raised.

Code:

```
import unittest
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.alert import Alert
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions

driver = webdriver.Chrome('./chromedriver')

def open_chat_app():
    driver.get("http://localhost/chat/")
    driver.implicitly_wait(20)

class Test(unittest.TestCase):
    def test_title(self):
        open_chat_app()
        self.assertEqual("Chat", driver.title, "Correct web page is not opened")

    def test_different_password(self):
        open_chat_app()
        signup_tab = driver.find_elements_by_class_name("opt")[1]

        signup_tab.click()

        signup_name = driver.find_element_by_id("signup_name")
        signup_username = driver.find_element_by_id("signup_username")
        signup_dob = driver.find_element_by_id("dob")
        signup_email = driver.find_element_by_id("signup_email")
        signup_password = driver.find_element_by_id("signup_password")
        signup_password2 = driver.find_element_by_id("signup_password2")
        signup_name.clear()
        signup_name.send_keys("Divyansh Tiwari")

        signup_username.clear()
        signup_username.send_keys("DragoDoom")

        signup_dob.clear()
        signup_dob.send_keys("03042000")

        signup_email.clear()
        signup_email.send_keys("inasumaeleven.gautam8@gmail.com")

        signup_password.clear()
        signup_password.send_keys("Divyansh@03")

        signup_password2.clear()
        signup_password2.send_keys("Divyansh@3")
        #search_bar.send_keys(Keys.RETURN)

        signup_submit = driver.find_element_by_id("signup_submit")
        signup_submit.click()

        # create alert object
        alert = Alert(driver)
        self.assertEqual("Passwords do not match!", alert.text, "Wrong password error is not raised")
        alert.accept()

    def test_wrong_email(self):
        open_chat_app()
        signup_tab = driver.find_elements_by_class_name("opt")[1]

        signup_tab.click()

        signup_name = driver.find_element_by_id("signup_name")
        signup_username = driver.find_element_by_id("signup_username")
        signup_dob = driver.find_element_by_id("dob")
        signup_email = driver.find_element_by_id("signup_email")
        signup_password = driver.find_element_by_id("signup_password")
        signup_password2 = driver.find_element_by_id("signup_password2")
        signup_name.clear()
        signup_name.send_keys("Divyansh Tiwari")

        signup_username.clear()
        signup_username.send_keys("DragoDoom")

        signup_dob.clear()
        signup_dob.send_keys("03042000")

        signup_email.clear()
        signup_email.send_keys("YE EMAIL GALAT HAI")

        signup_password.clear()
        signup_password.send_keys("Divyansh@03")

        signup_password2.clear()
        signup_password2.send_keys("Divyansh@03")
        #search_bar.send_keys(Keys.RETURN)

        signup_submit = driver.find_element_by_id("signup_submit")
        signup_submit.click()

        # create alert object
        alert = Alert(driver)
        self.assertEqual("Please enter a valid email address", alert.text, "Invalid Email error is not
raised")
        alert.accept()

    def test_empty_name(self):
        open_chat_app()
        signup_tab = driver.find_elements_by_class_name("opt")[1]

        signup_tab.click()

        signup_name = driver.find_element_by_id("signup_name")
        signup_username = driver.find_element_by_id("signup_username")
        signup_dob = driver.find_element_by_id("dob")
        signup_email = driver.find_element_by_id("signup_email")
        signup_password = driver.find_element_by_id("signup_password")
        signup_password2 = driver.find_element_by_id("signup_password2")
        signup_name.clear()
        signup_name.send_keys("")

        signup_username.clear()
        signup_username.send_keys("DragoDoom")

        signup_dob.clear()
        signup_dob.send_keys("03042000")

        signup_email.clear()
        signup_email.send_keys("inasumaeleven.gautam8@gmail.com")

        signup_password.clear()
        signup_password.send_keys("Divyansh@03")

        signup_password2.clear()
        signup_password2.send_keys("Divyansh@03")
        #search_bar.send_keys(Keys.RETURN)

        signup_submit = driver.find_element_by_id("signup_submit")
        signup_submit.click()

        # create alert object
        alert = Alert(driver)
        self.assertEqual("One or more required fields are empty", alert.text, "Empty Name Field error
is not raised")
        alert.accept()

    def test_empty_username(self):
        open_chat_app()
        signup_tab = driver.find_elements_by_class_name("opt")[1]

        signup_tab.click()

        signup_name = driver.find_element_by_id("signup_name")
        signup_username = driver.find_element_by_id("signup_username")
        signup_dob = driver.find_element_by_id("dob")
        signup_email = driver.find_element_by_id("signup_email")
        signup_password = driver.find_element_by_id("signup_password")
        signup_password2 = driver.find_element_by_id("signup_password2")

        signup_name.clear()
        signup_name.send_keys("Divyansh Tiwari")

        signup_username.clear()
        signup_username.send_keys("")

        signup_dob.clear()
        signup_dob.send_keys("03042000")

        signup_email.clear()
        signup_email.send_keys("inasumaeleven.gautam8@gmail.com")

        signup_password.clear()
        signup_password.send_keys("Divyansh@03")

        signup_password2.clear()
        signup_password2.send_keys("Divyansh@03")
        #search_bar.send_keys(Keys.RETURN)

        signup_submit = driver.find_element_by_id("signup_submit")
        signup_submit.click()

        # create alert object
        alert = Alert(driver)
        self.assertEqual("One or more required fields are empty", alert.text, "Empty username Field
error is not raised")
        alert.accept()

    def test_empty_email(self):
        open_chat_app()
        signup_tab = driver.find_elements_by_class_name("opt")[1]

        signup_tab.click()

        signup_name = driver.find_element_by_id("signup_name")
        signup_username = driver.find_element_by_id("signup_username")
        signup_dob = driver.find_element_by_id("dob")
        signup_email = driver.find_element_by_id("signup_email")
        signup_password = driver.find_element_by_id("signup_password")
        signup_password2 = driver.find_element_by_id("signup_password2")

        signup_name.clear()
        signup_name.send_keys("Divyansh Tiwari")

        signup_username.clear()
        signup_username.send_keys("DragoDoom")

        signup_dob.clear()
        signup_dob.send_keys("03042000")

        signup_email.clear()
        signup_email.send_keys("")

        signup_password.clear()
        signup_password.send_keys("Divyansh@03")

        signup_password2.clear()
        signup_password2.send_keys("Divyansh@03")
        #search_bar.send_keys(Keys.RETURN)

        signup_submit = driver.find_element_by_id("signup_submit")
        signup_submit.click()

        # create alert object
        alert = Alert(driver)
        self.assertEqual("One or more required fields are empty", alert.text, "Empty email Field error
is not raised")
        alert.accept()

    def test_empty_pass(self):
        open_chat_app()
        signup_tab = driver.find_elements_by_class_name("opt")[1]

        signup_tab.click()

        signup_name = driver.find_element_by_id("signup_name")
        signup_username = driver.find_element_by_id("signup_username")
        signup_dob = driver.find_element_by_id("dob")
        signup_email = driver.find_element_by_id("signup_email")
        signup_password = driver.find_element_by_id("signup_password")
        signup_password2 = driver.find_element_by_id("signup_password2")

        signup_name.clear()
        signup_name.send_keys("Divyansh Tiwari")

        signup_username.clear()
        signup_username.send_keys("DragoDoom")

        signup_dob.clear()
        signup_dob.send_keys("03042000")

        signup_email.clear()
        signup_email.send_keys("inasumaeleven.gautam8@gmail.com")

        signup_password.clear()
        signup_password.send_keys("")

        signup_password2.clear()
        signup_password2.send_keys("Divyansh@03")
        #search_bar.send_keys(Keys.RETURN)

        signup_submit = driver.find_element_by_id("signup_submit")
        signup_submit.click()

        # create alert object
        alert = Alert(driver)
        self.assertEqual("One or more required fields are empty", alert.text, "Empty PAssword Field
error is not raised")
        alert.accept()

    def test_empty_pass_2(self):
        open_chat_app()
        signup_tab = driver.find_elements_by_class_name("opt")[1]

        signup_tab.click()

        signup_name = driver.find_element_by_id("signup_name")
        signup_username = driver.find_element_by_id("signup_username")
        signup_dob = driver.find_element_by_id("dob")
        signup_email = driver.find_element_by_id("signup_email")
        signup_password = driver.find_element_by_id("signup_password")
        signup_password2 = driver.find_element_by_id("signup_password2")

        signup_name.clear()
        signup_name.send_keys("Divyansh Tiwari")

        signup_username.clear()
        signup_username.send_keys("DragoDoom")

        signup_dob.clear()
        signup_dob.send_keys("03042000")

        signup_email.clear()
        signup_email.send_keys("inasumaeleven.gautam8@gmail.com")

        signup_password.clear()
        signup_password.send_keys("Divyansh@03")

        signup_password2.clear()
        signup_password2.send_keys("")
        #search_bar.send_keys(Keys.RETURN)

        signup_submit = driver.find_element_by_id("signup_submit")
        signup_submit.click()

        # create alert object
        alert = Alert(driver)
        self.assertEqual("One or more required fields are empty", alert.text, "Empty Confirm
Password Field error is not raised")
        alert.accept()

    def test_signup(self):
        open_chat_app()
        signup_tab = driver.find_elements_by_class_name("opt")[1]

        signup_tab.click()

        signup_name = driver.find_element_by_id("signup_name")
        signup_username = driver.find_element_by_id("signup_username")
        signup_dob = driver.find_element_by_id("dob")
        signup_email = driver.find_element_by_id("signup_email")
        signup_password = driver.find_element_by_id("signup_password")
        signup_password2 = driver.find_element_by_id("signup_password2")

        signup_name.clear()
        signup_name.send_keys("Divyansh Tiwari")

        signup_username.clear()
        signup_username.send_keys("Drago-Doom")

        signup_dob.clear()
        signup_dob.send_keys("03042000")

        signup_email.clear()
        signup_email.send_keys("inasumaeleven.gautam9@gmail.com")

        signup_password.clear()
        signup_password.send_keys("Divyansh@03")

        signup_password2.clear()
        signup_password2.send_keys("Divyansh@03")
        #search_bar.send_keys(Keys.RETURN)

        signup_submit = driver.find_element_by_id("signup_submit")
        signup_submit.click()

        WebDriverWait(driver,10).until(expected_conditions.alert_is_present())

        # create alert object
        alert = Alert(driver)
        self.assertEqual("Account Successfully Created", alert.text, "Account is not being created")
        ale = driver.switch_to_alert();
        alert.accept()
        ale.accept()

if __name__ == '__main__':
    unittest.main()
```

Output:



```
Traceback (most recent call last):
  File "C:\Users\inasu\Desktop\SQAT\SignUp_Testing.py", line 16, in test_title
    open_chat_app()
  File "C:\Users\inasu\Desktop\SQAT\SignUp_Testing.py", line 11, in open_chat_app
    driver.get("http://localhost/chat/")
  File "C:\Users\inasu\AppData\Local\Programs\Python\Python37\lib\site-packages\selenium\webdriver\remote\webdriver.py", line 430, in get
    self.execute(Command.GET, {'url': url})
  File "C:\Users\inasu\AppData\Local\Programs\Python\Python37\lib\site-packages\selenium\webdriver\remote\webdriver.py", line 418, in execute
    self.error_handler.check_response(response)
  File "C:\Users\inasu\AppData\Local\Programs\Python\Python37\lib\site-packages\selenium\webdriver\remote\errorhandler.py", line 242, in check_response
    raise exception_class(message, screen, stacktrace, alert_text)  # type: ignore[call-arg]  # mypy is not smart enough here
selenium.common.exceptions.UnexpectedAlertPresentException: Alert Text: Account Successfully Created
Message: unexpected alert open: {Alert text : Account Successfully Created}
  (Session info: chrome=94.0.4606.81)
Stacktrace:
Backtrace:
	Ordinal0 [0x011ABDE3+2473443]
	Ordinal0 [0x01146661+2057825]
	Ordinal0 [0x01052438+1057848]
	Ordinal0 [0x010A48FB+1394939]
	Ordinal0 [0x010955AB+1332651]
	Ordinal0 [0x01072104+1188100]
	Ordinal0 [0x01072F59+1191769]
	GetHandleVerifier [0x01332266+1549718]
	GetHandleVerifier [0x013DD4A7+2250711]
	GetHandleVerifier [0x0123718B+521403]
	GetHandleVerifier [0x01236229+517465]
	Ordinal0 [0x0114B79D+2078621]
	Ordinal0 [0x0114FB58+2095960]
	Ordinal0 [0x0114FC92+2096274]
	Ordinal0 [0x01159541+2135361]
	BaseThreadInitThunk [0x77236739+25]
	RtlGetFullPathName_UEx [0x77468AFF+1215]
	RtlGetFullPathName_UEx [0x77468ACD+1165]


----------------------------------------------------------------------
Ran 9 tests in 8.880s

FAILED (errors=2)
>>>
```

Module 2: Login

For the login Module, we again need to check appropriate errors are being raised id invalid details are being entered

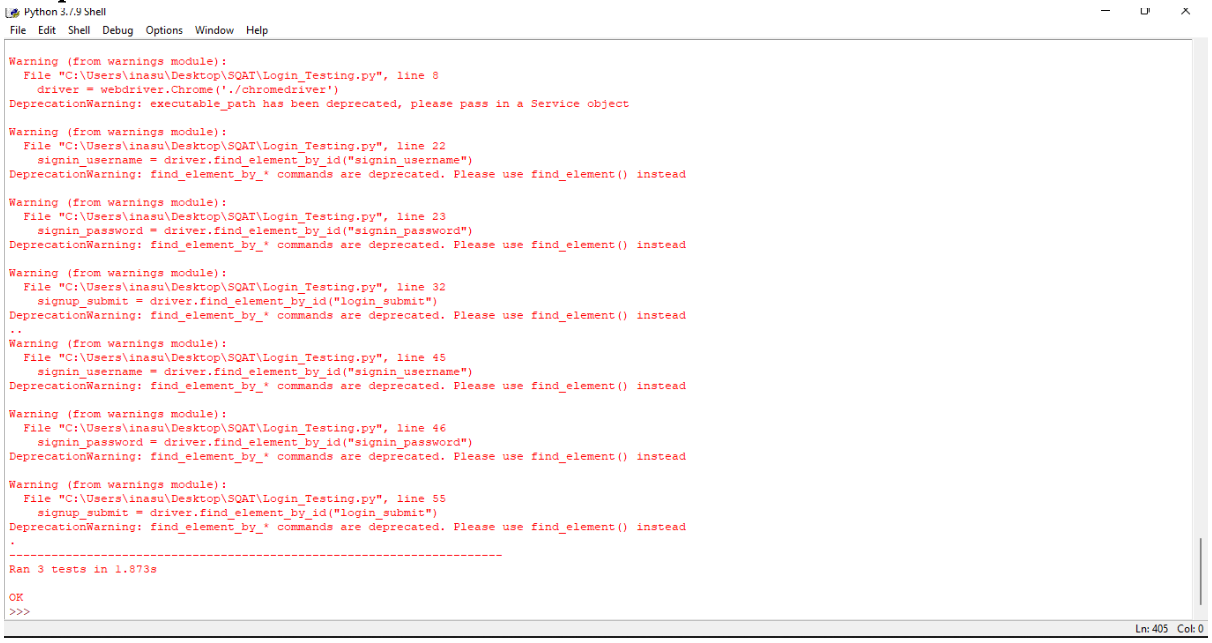
Code:

```
import unittest
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.alert import Alert
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions

driver = webdriver.Chrome('./chromedriver')

def open_chat_app():
    driver.get("http://localhost/chat/")
    driver.implicitly_wait(20)

class Test(unittest.TestCase):
    def test_title(self):
        open_chat_app()
        self.assertEqual("Chat", driver.title, "Correct web page is not opened")
```

```
def test_different_password(self):
    open_chat_app()

    signin_username = driver.find_element_by_id("signin_username")
    signin_password = driver.find_element_by_id("signin_password")

    signin_username.clear()
    signin_username.send_keys("Drago-Doom")

    signin_password.clear()
    signin_password.send_keys("Divyansh@0345")
    #search_bar.send_keys(Keys.RETURN)

    signup_submit = driver.find_element_by_id("login_submit")
    signup_submit.click()

    WebDriverWait(driver,10).until(expected_conditions.alert_is_present())

    # create alert object
    alert = Alert(driver)
    self.assertEqual("Invalid Login Credentials", alert.text, "Wrong creds error is not raised")
    alert.accept()

def test_wrong_username(self):
    open_chat_app()

    signin_username = driver.find_element_by_id("signin_username")
    signin_password = driver.find_element_by_id("signin_password")

    signin_username.clear()
    signin_username.send_keys("DragDoom")

    signin_password.clear()
    signin_password.send_keys("Divyansh@0345")
    #search_bar.send_keys(Keys.RETURN)

    signup_submit = driver.find_element_by_id("login_submit")
    signup_submit.click()

    WebDriverWait(driver,10).until(expected_conditions.alert_is_present())
```

```
        # create alert object
        alert = Alert(driver)
        self.assertEqual("Invalid Login Credentials", alert.text, "Wrong creds error is not raised")
        alert.accept()

if __name__ == '__main__':
    unittest.main()
```

Output:



```
Warning (from warnings module):
  File "C:\Users\inasu\Desktop\SQAT\Login_Testing.py", line 8
    driver = webdriver.Chrome('./chromedriver')
DeprecationWarning: executable_path has been deprecated, please pass in a Service object

Warning (from warnings module):
  File "C:\Users\inasu\Desktop\SQAT\Login_Testing.py", line 22
    signin_username = driver.find_element_by_id("signin_username")
DeprecationWarning: find_element_by_* commands are deprecated. Please use find_element() instead

Warning (from warnings module):
  File "C:\Users\inasu\Desktop\SQAT\Login_Testing.py", line 23
    signin_password = driver.find_element_by_id("signin_password")
DeprecationWarning: find_element_by_* commands are deprecated. Please use find_element() instead

Warning (from warnings module):
  File "C:\Users\inasu\Desktop\SQAT\Login_Testing.py", line 32
    signup_submit = driver.find_element_by_id("login_submit")
DeprecationWarning: find_element_by_* commands are deprecated. Please use find_element() instead
..
Warning (from warnings module):
  File "C:\Users\inasu\Desktop\SQAT\Login_Testing.py", line 45
    signin_username = driver.find_element_by_id("signin_username")
DeprecationWarning: find_element_by_* commands are deprecated. Please use find_element() instead

Warning (from warnings module):
  File "C:\Users\inasu\Desktop\SQAT\Login_Testing.py", line 46
    signin_password = driver.find_element_by_id("signin_password")
DeprecationWarning: find_element_by_* commands are deprecated. Please use find_element() instead

Warning (from warnings module):
  File "C:\Users\inasu\Desktop\SQAT\Login_Testing.py", line 55
    signup_submit = driver.find_element_by_id("login_submit")
DeprecationWarning: find_element_by_* commands are deprecated. Please use find_element() instead
.
----------------------------------------------------------------------
Ran 3 tests in 1.873s

OK
>>>
```

The entire Demonstration of the Selenium Testing can be checked here:

.mp4

video2999452637.mp4

CONCLUSION

We've successfully performed Testing on our Chatting Website. We've checked for Coding and Formatting errors using TIDY and we've performed Unit Testing using Python – Unit test module and Selenium Web Driver for Browser Automation.

There were no major flaws in the code. However the code was optimized wherever needed.